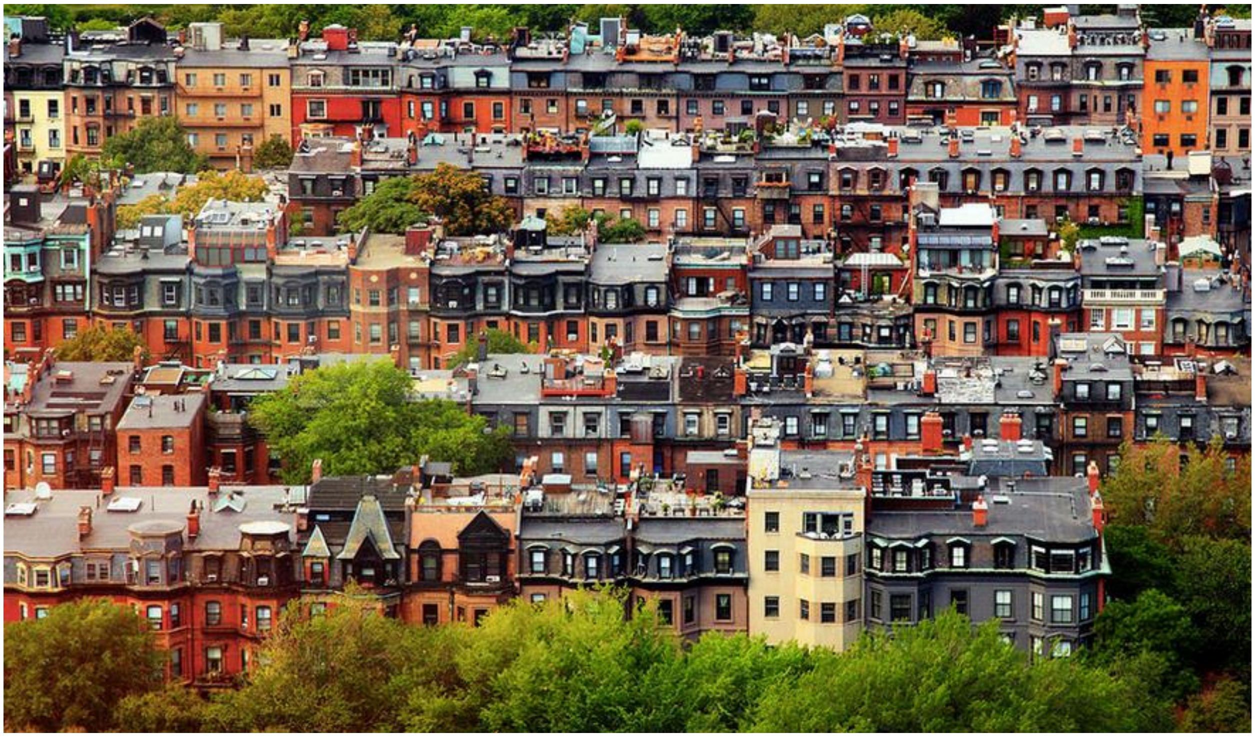
House Price Prediction



First Name: Vaidiyanathan

Last Name: Lalgudi Venkatesan

Executive Summary

Executive Summary:

The objective is to analyze the Boston housing dataset and to predict the medv variable. Different models were fit and compared using the model MSE and out of sample prediction accuracy was compared for the models.

PROCEDURE:

- The data was randomly sampled into training (75%) and testing data (25%).
- Then, linear regression model was fit on the training set
- Further, different modeling techniques such as Decision trees, Bagging, Random Forests and boosting were used to model the training data.
- Finally, Gam and Neural network models were fit
- In sample MSE was used to identify the best fit model
- Out of sample was also calculated for the same to give a good comparison in model performance on the test data

OBSERVATIONS:

The results of different model MSE and test MSE is shown below:

Model	Train MSE	Test MSE
Linear Regression	0.355	28.23
Stepwise	20.77	28.05
Decision Tree	15.05	25.01
Bagging	11.51	14.75
Random Forest	10.64	10.93
Boosting	0.018	10.81
GAM	7.190	16.738
Neural Network	0.002	70.36

INFERENCE:

The in-sample prediction is less that the out of sample as the model tries to fit the training data in all models. But we find that the model MSE is low in random forest and almost zero in boosting. Also, we see that the Random forest and boosting perform best with respect to out of sample prediction.

BOSTON HOUSING DATA

1.1 Linear Regression Model:

On modeling the data using linear regression, including all the variables, we observed the linear model given by the following equation:

*Medv ~ 36.71 – 0.09*crim + 0.05*zn + 1.95*chas – 19.2*nox + 3.83*rm- 1.48*dis + 0.29*rad – 0.01*tax – 0.98*ptratio + 0.01*black – 0.55*lstat*

We find that the variables “indus” and “age” are found to have p-value> alpha and so are not significantly impacting the prediction value of medv. Thus these two variables can be ignored while generating a model to predict the medv variable.

In sample MSE	: 20.84358
Out of sample MSE	: 28.23323

1.2 Step Wise Selection

Next, we wanted to perform stepwise variable selection for the model that predicts the medv value based on all other variables. The beta coefficients for the resulting model is shown below

*Medv ~ 36.17 - 0.54*lstat + 3.84*rm - 0.96*ptratio + 0.01*black - 1.55*dis – 17.79*nox + 2.04*chas + 0.04*zn + 0.27*rad – 0.01*tax – 0.1*crim*

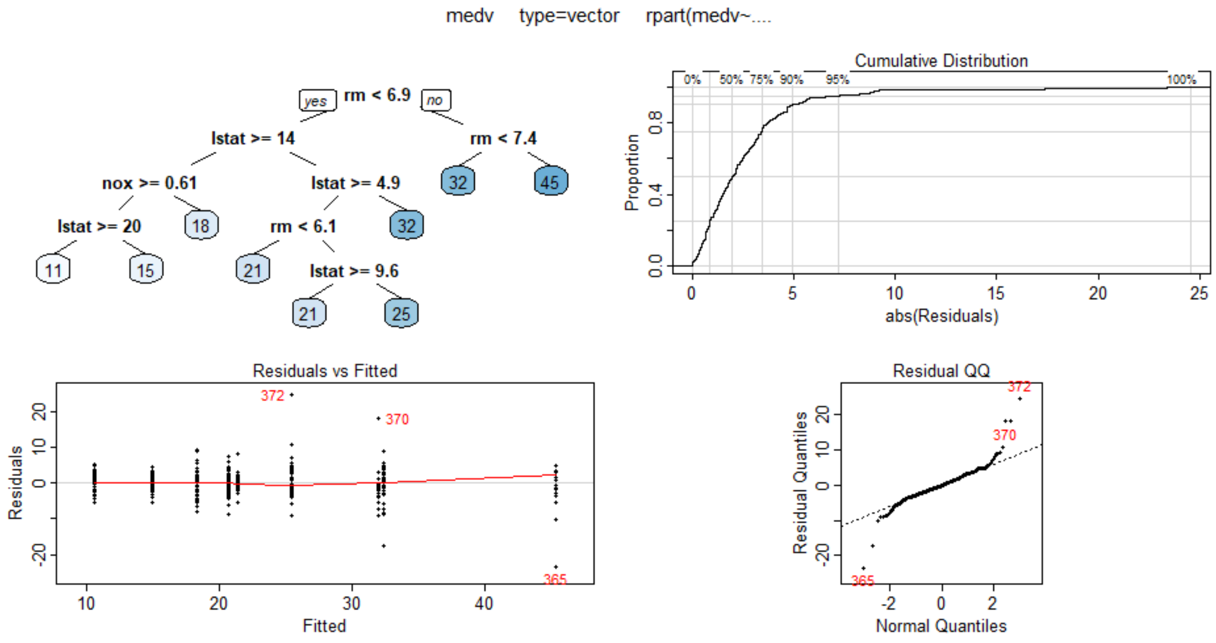
We find that the model with all variables except the “indus” and “age” variables provides the least combination AIC values. This is consistent with the result we obtained by performing a simple linear regression model in the previous section.

In sample MSE	: 20.77397
Out of sample MSE	: 28.05547

1.3 CART Based model – Decision Tree:

We then wanted to perform the analysis by developing a CART model using the dataset. We get the following tree from the analysis as shown in [Figure-9]. We also obtain the following residual diagnostic for the CART model developed. We find that the distribution of the predicted variable is almost normal but is skewed at the tails. An introduction of a quadratic term might be used to subside this effect.

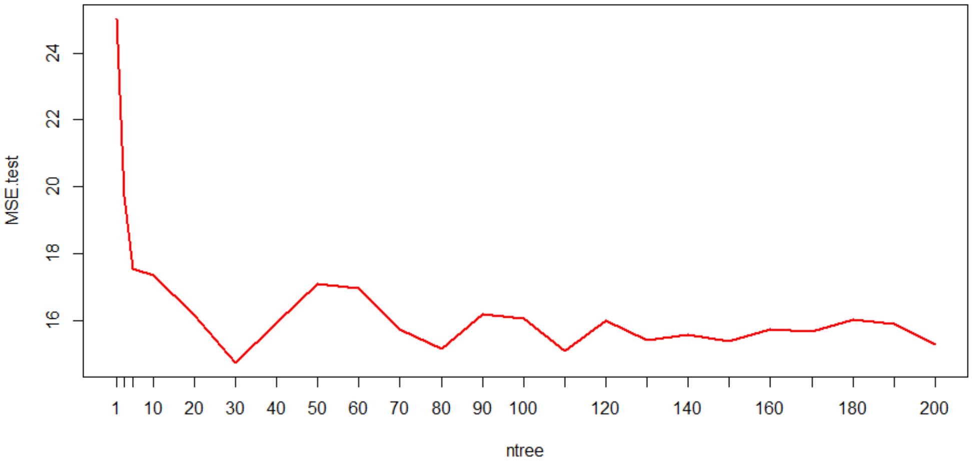
In sample MSE	: 15.05392
Out of sample MSE	: 25.01981



[Figure 9]

1.4 Bagging:

Bagging was performed using 100 bootstrap samples to fit the model using ensemble method. A Grid search on the optimal number of trees required to reduce the MSE was done and the number of trees were identified as 30.

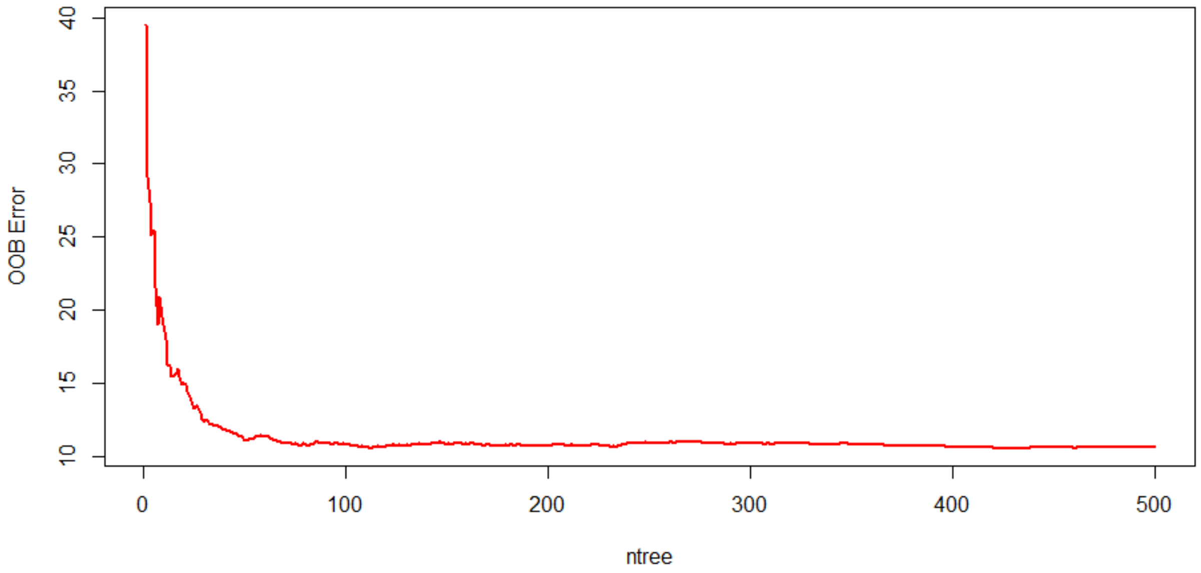


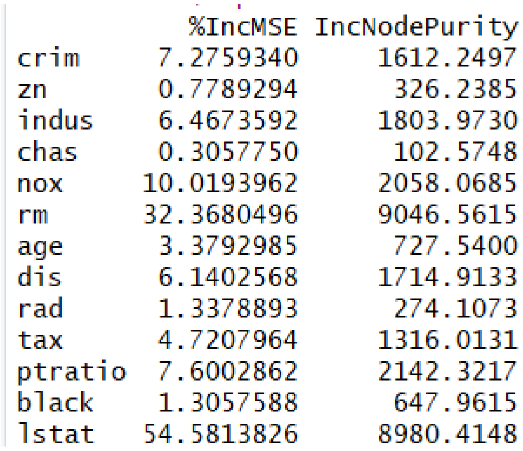
In sample MSE : 11.51375

Out of sample MSE : 14.75407

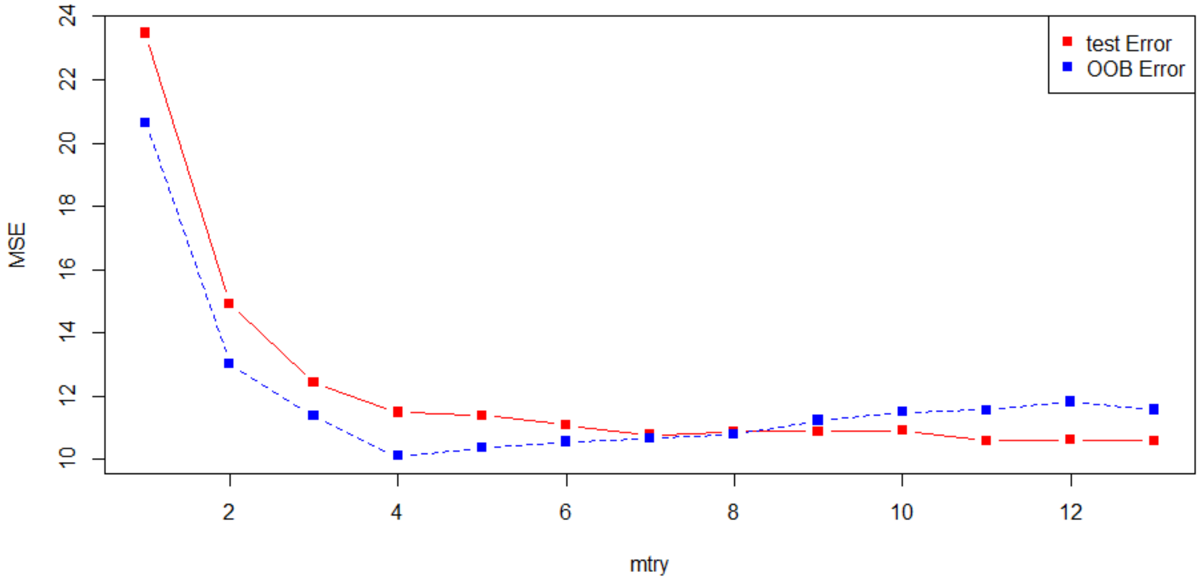
1.5 Random Forest:

Random forest was then used to model the training data. The Out of bag error for increasing trees was noted. Also, the hyperparameter tuning for the number of predictors to be selected, that is, mtry was identified using grid search. This gave improved results and the variable importance is as below.





```
           %IncMSE IncNodePurity
crim     7.2759340     1612.2497
zn       0.7789294      326.2385
indus    6.4673592     1803.9730
chas     0.3057750      102.5748
nox     10.0193962     2058.0685
rm      32.3680496     9046.5615
age      3.3792985      727.5400
dis      6.1402568     1714.9133
rad      1.3378893      274.1073
tax      4.7207964     1316.0131
ptratio  7.6002862     2142.3217
black    1.3057588      647.9615
lstat   54.5813826     8980.4148
```

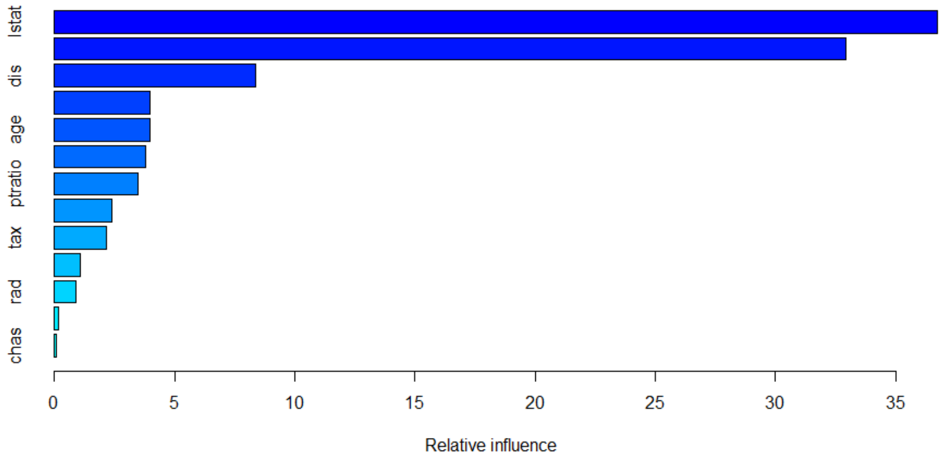


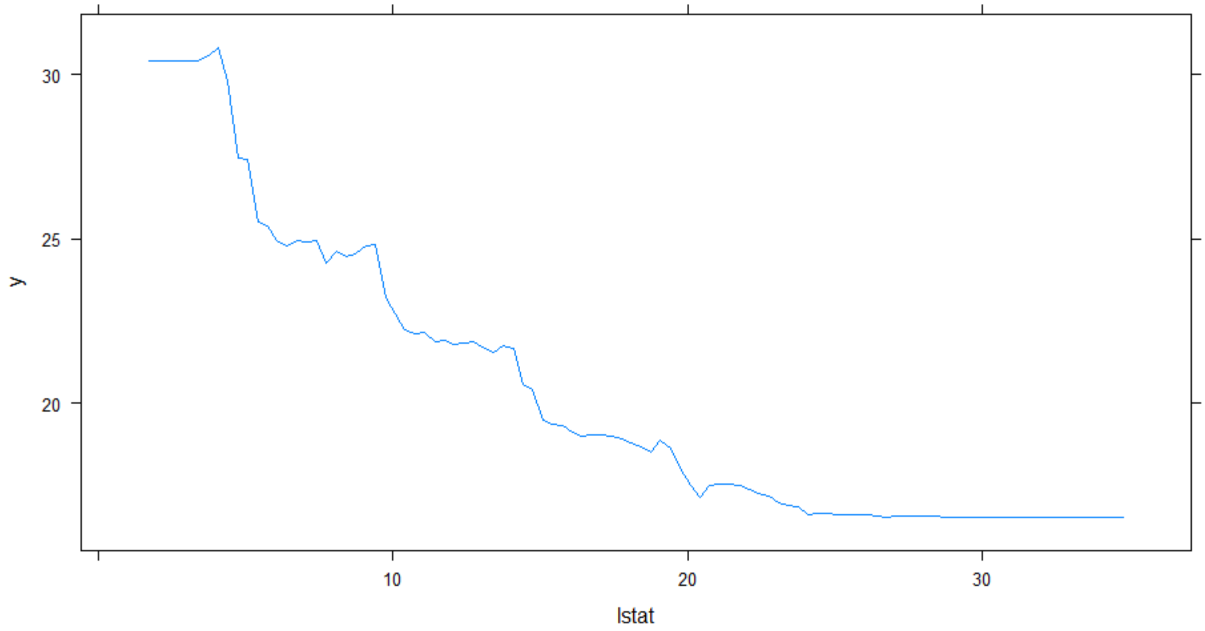
It was found that using 11 predictors gave the least test error and hence it was used for the model

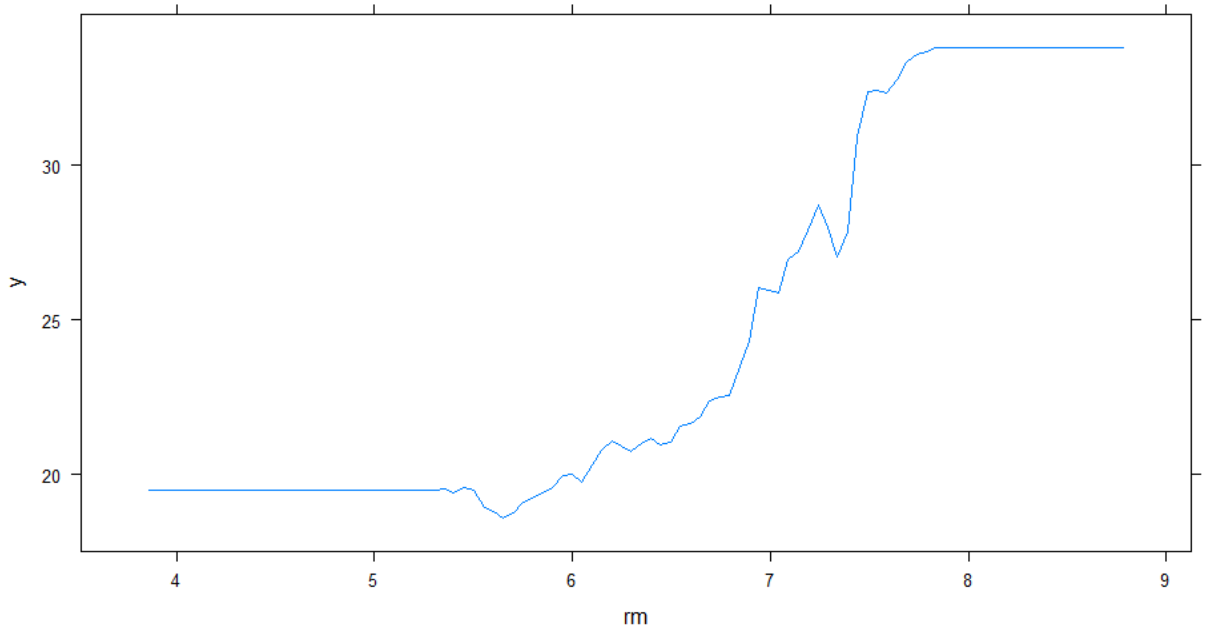
In sample MSE	: 10.6484
Out of sample MSE	: 10.932

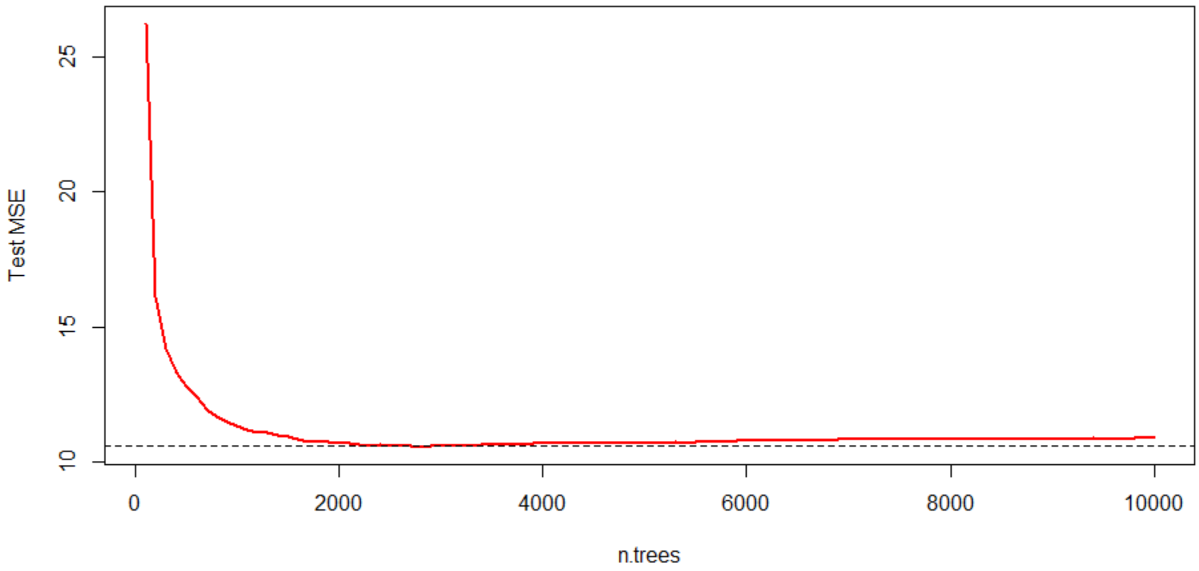
1.6 Boosting:

Finally, boosting was performed to model the training data with 10000 trees and a learning rate of .01 and interaction depth of 8 variables.




30
25
20
y
10
20
30
lstat


30
25
20
y
4
5
6
7
8
9
rm



We can also see the performance of boosting as the number of trees increase, we see that the MSE decreases.

In sample MSE	: 0.01857575
Out of sample MSE	: 10.81737

1.7 Generalized Additive Models (GAM):

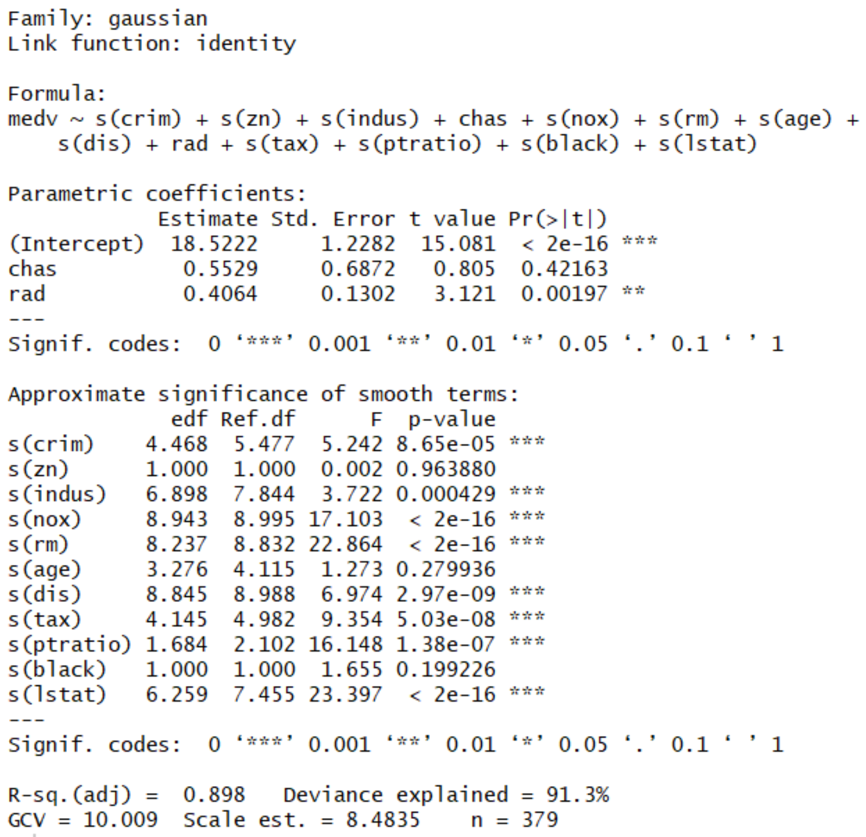
The next approach was to try a GAM model to predict the medv variable. We get the following coefficients as shown below.

We find that the variables Zn, Age and Black doesn’t seem to have a significant impact on the medv variable.

Model Performance:

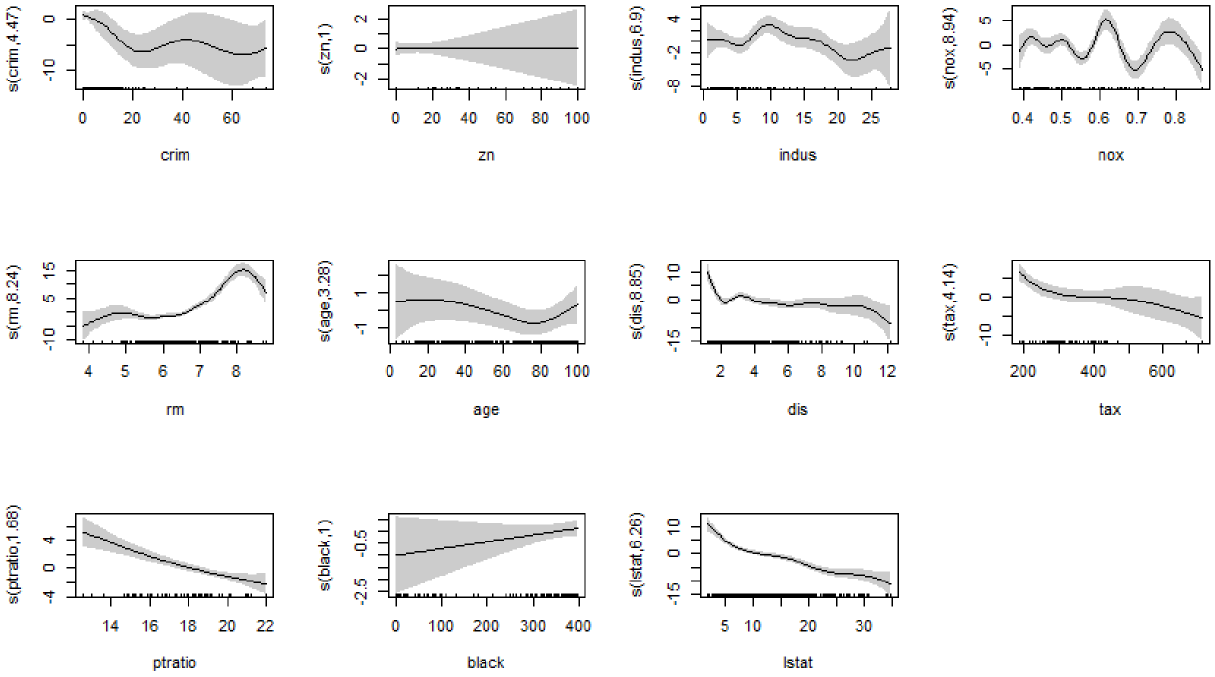
In sample MSE	: 7.190
Out of sample MSE	: 16.738

As expected we find that the MSE for the training data is lesser than the MSE of the test data. Also, we find that the error is lesser in the GAM model than the GLM and tree models.



```
Family: gaussian
Link function: identity

Formula:
medv ~ s(crim) + s(zn) + s(indus) + chas + s(nox) + s(rm) + s(age) +
    s(dis) + rad + s(tax) + s(ptratio) + s(black) + s(lstat)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  18.5222     1.2282  15.081  < 2e-16 ***
chas          0.5529     0.6872   0.805  0.42163    
rad           0.4064     0.1302   3.121  0.00197 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
             edf Ref.df      F  p-value    
s(crim)    4.468  5.477  5.242 8.65e-05 ***
s(zn)      1.000  1.000  0.002 0.963880    
s(indus)   6.898  7.844  3.722 0.000429 ***
s(nox)     8.943  8.995 17.103  < 2e-16 ***
s(rm)      8.237  8.832 22.864  < 2e-16 ***
s(age)     3.276  4.115  1.273 0.279936    
s(dis)     8.845  8.988  6.974 2.97e-09 ***
s(tax)     4.145  4.982  9.354 5.03e-08 ***
s(ptratio) 1.684  2.102 16.148 1.38e-07 ***
s(black)   1.000  1.000  1.655 0.199226    
s(lstat)   6.259  7.455 23.397  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.898   Deviance explained = 91.3%
GCV = 10.009  Scale est. = 8.4835    n = 379
```

Also, we find that few variables are non linear such as indus, while few are linear such as lstat variables.

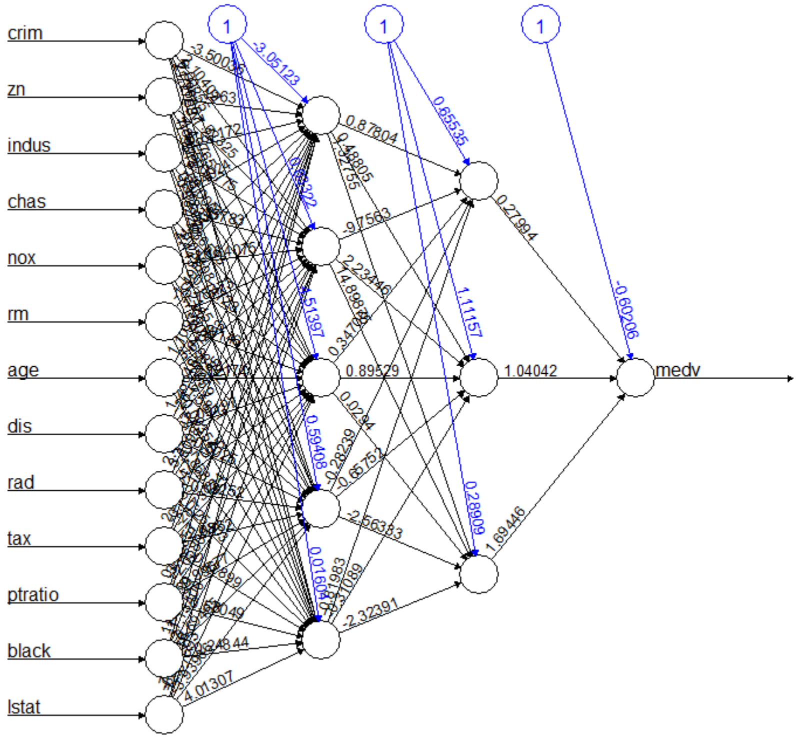


1.8 Neural Network:

The final approach that was tried was to construct a neural network to predict the medv variable. There were several steps involved in implementing the neural network model for the prediction.

Firstly, all the predictor variables were standardized to (0 to1) range by using the min and max values of the corresponding variables.

We get the following model with 2 hidden layers:



Model Performance:

In sample MSE : 0.002
Out of sample MSE : 70.36

We find that the training MSE (seed of 12969599) comes very low, but the test MSE is very high. It overfits the training data.